東海大學電機工程學研究所

碩士學位論文

LoRaWAN 物聯網之低功耗資料加密方法

Low Power Data Encryption Method For LoRaWAN-based IoT

指導教授：蔡坤霖 博士

研 究 生：黃郁淩

中華民國 108 年 1 月

致謝

這份論文得以完成，首先要感謝我的指導教授蔡坤霖老師，老師耐心、無私的教導，以及彈性的學習方式，使我在研究所期間獲得無法量化的成長。謝謝老師給我機會能發掘適合的領域，在此致上最深的感謝及敬意。

感謝口試委員：張延任博士、林嬿雯博士、陳弘明博士，在口試時給我許多論文上的實質建議。

感謝羣倫學長與嘉廷前輩，在我就學的期間，給予極大的彈性及協助。耐心的指導及培養我的實務經驗。謝謝學長益豪與鉦新，在我工作與學業來回奔波時，給予幫助及包容。

最後要感謝我的家人，在研究所期間給予的協助及支持，讓我能順利的完成研究所的學業。

摘要

近年來，物聯網（Internet of Things, IoT）不只侷限於工業用途，也應用在許多截然不同的領域，例如健康照護與智慧農場。物聯網漸漸的成為熱門的基礎建設，大量的資料收集以供後續的決策依據，科技正逐漸的提升我們的生活環境。

LoRaWAN為其中一個用於IoT資料傳輸的低功耗廣域網路協議，由LoRa聯盟提出，低功率且長距離的特色使其備受矚目。LoRaWAN採用Advanced Encryption Standard（AES）加密方法，AES使用代數運算及多重加密循環來確保其通訊安全。LoRaWAN透過為不同的終端裝置設置相異的傳輸延遲來降低其通訊功率，但是，AES加密終端裝置的功耗並未被考慮到。

本論文提出一種高安全性且低功耗的通訊方案，稱為低功耗資料加密方法(Low Power Data Encryption Method，LPDEM) ，旨在透過減少AES的加密循環，降低終端裝置的數據加密功耗。在LPDEM中，定期更新加密金鑰與加入D-Box更新程序以增強安全等級，並簡化AES的加密過程，最終進一步降低功耗但仍具備高安全性。與傳統AES相比，分析結果指出LPDEM可以將加密功率降至26.2％。LPDEM可以有效抵禦三種攻擊，包括已知密鑰，重送和竊聽攻擊，並且適合使用於LoRaWAN的IoT環境中。

Abstract

In recent years, Internet of Things (IoT) has been used not only for industria, but also in many distinct domains, such as healthcare and smart farming. Internet of Things has gradually become a popular infrastructure. Technology is helping people to do something, aiming to improve our living environments.

LoRaWAN, developed by LoRa Alliance, is one of the Low Power Wide Area Network (LPWAN) protocols. Its low-power and long-distance features make it suitable for IoT environments. LoRaWAN adopts Advanced Encryption Standard (AES) for data encryption, which uses algebraic operations and multiple encryption cycles to ensure its communication security. LoRaWAN reduces communication power by setting different transmission latencies for different end-devices. However, AES does not take into account its end device's encryption power.

This research proposes a high-security and low-power communication scheme for LoRaWAN, named Low Power Data Encryption Method (LPDEM), is aimed to reduce the power consumption of end-device's data encryption by reducing the encryption cycle of AES. In the LPDEM, encryption key and D-Box update procedure is presented to enhance the security level and simplify the AES encryption process so that the power consumption can be further lowered. Ultimately reducing power consumption but still has high security. Comparing with traditional AES, the analysis results show that the LPDEM can minimize the encryption power up to 26.2%. The LPDEM can also effectively resist three attacks, including known-key, replay and eavesdropping attacks, and is practically helpful for use in LoRaWAN-based IoT environments.

目錄

圖目錄

表目錄

第一章 緒論

1.1 研究背景

在經過一番摸索與墾荒期之後，物聯網（Internet of Things, IoT）已進入快速發展的階段，日新月異的通訊革新，各式基於物聯網的應用隨之衍生，例如環境監控[1] [2]、智慧工廠、智慧家居[3] [4]、醫療及公共衛生物聯網[5]、智能農場[6]、智能交通運輸系統[7]……，根據Ericsson Mobility Report 2017 [8]，在2022年將有一百八十億多個的物聯網裝置連上網。

當今的物聯網（IoT）結合人工智慧（AI）與深度學習，隨著終端裝置越來越密切的互連，為AI提供了廣大的基礎。匯流成新應用智慧物聯AIoT，透過物聯網感知整個環境，讓生活中的每個場景都數位化，經過包括5G在內的各種互連，再將這些資料經過AI處理，實現革命性變化。物聯網概念的應用越來越深入生活的大小物，例如自動駕駛[9]、車聯網[10]。部署的範圍也越來越廣，2017年，阿里巴巴在無錫鴻山打造了中國第一個物聯網小鎮[11]，小鎮中部署了大量的感測器，再透過LoRa協議，把資料擷取到阿里雲平台上統一管理。萬物物聯網將用技術重新定義人類生活。

目前物聯網的資料傳輸協定主為常規的無線通訊技術，例如：4G, Wi-FI, Bluetooth, Zigbee 或是傳統乙太網路。然而，隨著物聯網應用發展，具有長距離、高穩定性，低功耗、低資料量特色的通訊技術更受到關注。

越來越多企業或組織開始投入低功耗廣域網路(Low Power Wide Area Network, LPWAN)的建置，目前已提出並用於IoT資料傳輸的低功耗廣域網路協議包括Narrow Band IoT (NB-IoT) [12]、LoRaWAN[13]、Sigfox[14]、Weightless[15]、HaLow[16]及RPMA[17]。其中以NB-IoT與LoRaWAN最備受關注。

由3rd Generation Partnership Project (3GPP) 標準化的NB-IoT使用的是窄帶無線電技術，針對IoT打造的電信級網路，針對網路傳輸品質、數據安全有較高的保證，

再加上建置成本較低，企業不用大幅更改現行的4G LTE電信網路架構，能快速部署、允許連接大量的物聯網裝置、低成本及長待機時間，因而備受各國電信商所支持。

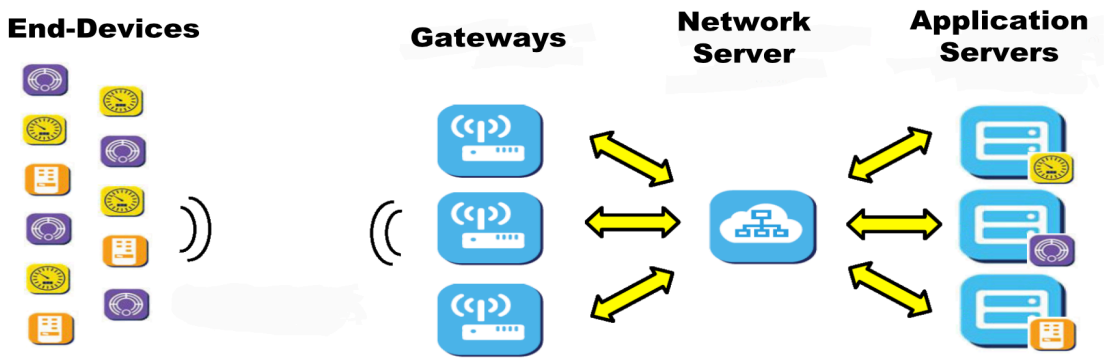
不過由於NB-IoT採用電信級的網路，適合高價貨物的追蹤，或是有數據精準度、及時性要求的物聯網。例如，與消費者有直接關聯的健康照護應用。工業控制、智慧製造、智慧農業等就不太適合。

LoRaWAN則是由LoRa聯盟開發，它支援長距離通訊，特定的頻寬、電池待命時間長、高品質及高附載的網路，以及較高的安全性。其中較突出的特點為：第一，長壽命。LoRaWAN網路非常省電，適合運用在不方便或久久換一次電池的情況下，例如在水表電池上，大概能有十年的壽命。第二，長距離。LoRaWAN支援長距離資料傳輸，過去資料無線的傳輸模式以Wi-Fi、藍牙等短距離為主，約在幾十米至幾百米範圍，LoRa則可達十幾公里的距離傳輸，適合大廠區的佈設。第三，低成本。LoRaWAN網路整個架構非常簡單，使用成本非常低。

1.2 LoRaWAN 簡介

LoRaWAN 具遠程通訊、低功耗及低成本的特色，根據 LoRa 聯盟規範定義的最長通訊距離可達十五至二十公里，因此在 LoRaWAN 的環境底下只需要少量的通訊閘。LoRaWAN 的特性在於電池壽命長，不傳輸訊息時裝置會進入深度睡眠模式，大幅延長充電週期。低功耗的特性延伸了更長的電池生命週期及其非授權頻段減少了設備的使用成本。

圖一顯示了 LoRaWAN 的網路脈絡，包含終端裝置（End-Device）、通訊閘道（Gateways）、網路服務器（Network Server）及應用層服務器（Application Server），終端裝置可以是感測器、儀表、監視器、控制器、設備.....等。

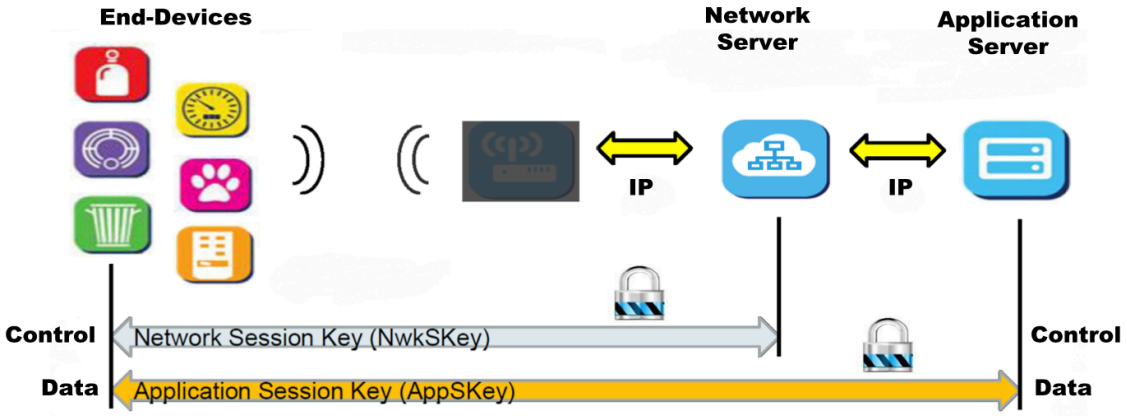


圖一 LoRaWAN網路架構示意圖 (資料來源:LoRa Alliance)

LoRa使用如Wi-Fi透過設置基地台（Wi-Fi access point）來建置網路環境的模式。通訊閘道透過乙太網路、3G/4G網路或Wi-Fi傳送來自終端裝置及網路服務器之間來回的訊息，網路服務器會檢查訊息完整性再將這些訊息傳輸給應用層服務器。應用層服務器解密訊息後根據訊息內容做相對應的決策與回應。

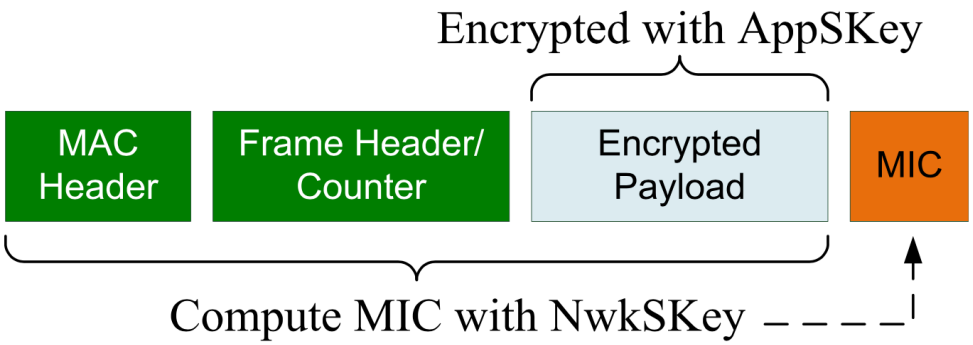
目前有些物聯網只針對通訊閘及服務器之間的數據傳輸做加密動作，但不對終端設備加密以節省終端設備的能源。然而，LoRaWan則提供終端加密機制，如圖二

所示，LoRaWAN透過Network Session Key (NwkSKey)替終端設備對網路服務器的數據加密、利用Application Session Key (AppSKey) 加密終端設備-應用服務器的數據傳輸。



圖二 LoRaWAN透過兩組金鑰達到點對點之安全保護 [13]

每一個訊息透過AES(Advanced Encryption Standard)加密演算法[18]產生相對應的MIC (Message Integrity Code)，並加在訊息的後方以便驗證。圖三說明了LoRaWAN的資料加密結構，包含MAC header、Frame header 、加密負載及由自身AES加密出來的Message Integrity Code (MIC)，其中MIC附加在訊息最尾端已用於檢查訊息完整性，一旦訊息內容遭竄改或是由非預設端點的虛造設備傳送，由網路服務器自身計算出的MIC將不會與接收到的MIC相等，因此可確保資料的完整性，並確認封包是否由合法的端點所傳遞。



圖三 LoRaWAN封包結構示意圖 [13]

此外，終端設備及應用層服務器之間的數據安全由終端設備利用 128 位元 AES 的 AppSKey 加密明文，生成帶有密文的加密負載，應用服務器使用一樣的 AppSKey 解密密文，NwkSKey 裝載在 LoRaWAN 網路，是為了驗證資料封包的真實性和完整性。每一對終端裝置與網路服務器（或應用服務器）都擁有獨一無二的 NwkSKey（或 AppSKey），兩個 Session Key (AppSKey and NwkSKey)相互認證，一個用來提供點對點應用程式(AppSKey)的加密，另一個則用來提供完整性保護。通訊閘、網路服務器、應用服務器的資料傳輸也利用 TLS (Transport Layer Security) 協定保護。

儘管LoRaWAN在資料傳輸時擁有較好的通訊方法可降低系統整體網路的功耗，且金鑰的部分也有良好的規範，但在LoRaWAN規劃的十年電池使用時間上，仍有許多可以改善討論的地方。

1.3 研究動機

在資訊安全的部分，LoRaWAN使用Advanced Encryption Standard (AES) [18]加密方式，雙向的身份認證、完整性檢查及資料加密特色保證其終端對終端的安全。終端裝置與網路服務器的雙向身份認證確保只有在通過認證的的設備可以連接LoRaWAN，意味著竊聽攻擊及無效裝置無法成功地通過驗證。

在過去的幾十年中，許多研究人員專注於物聯網資訊安全問題。Chahid 等人在文獻中[19]提出了物聯網安全問題並討論了許多解決方案。使用現代加密方法可以

保護數據完整性和通信安全[20]，然而，複雜加密步驟也浪費了大量的能量[21]。一般而言，物聯網中的終端設備通常預設使用輕型的電池運行並僅有有限的能源、存儲容量和處理能力。因此，在過去幾年中，許多研究[22][23]已經提出了各種方案來最小化終端設備的物聯網功耗。然而，考慮到無線感測網路的功耗和安全性，相關研究可以再進一步改善[24][25]。

為了平衡功耗和安全性，在本論文中希望為LoRaWAN提出了一種安全但低功耗的通訊方案Low Power Data Encryption Method（以下簡稱LPDEM），透過簡化加密過程來降低其終端設備消耗的數據加密功耗並使用動態加密密鑰以及查表用於增強通訊安全性，預期能夠抵抗已知密鑰攻擊[26]，重送攻擊[27]和竊聽攻擊[28]。

1.4 論文架構

本論文章節安排如下：

第一章為緒論，分別為「研究背景」、「研究動機與目的」及「論文架構」。第二章是相關研究與文獻探討，分別針對「物聯網安全性問題」、「LoRaWAN性能與安全性」、「物聯網功耗問題」及「AES-128加密法」等議題，討論物聯網與LoRaWAN的安全性與其功耗問題，最後說明所使用的加密方法AES-128與改善方向。第三章是「低功耗資料加密方法」，分別是「金鑰擴展」、「資料編譯」；在「金鑰擴展」說明如何提升DASS的計算，以及如何生成D-Box取代舊有S-Box，最後是Appskey與D-Box的更新程序說明；在「資料編譯」說明整個AES-128加密法的加密過程。第四章是安全性與功耗分析，分別是「安全性評估」與「功耗分析」；「安全性評估」分析LPDEM的安全性以及如何保護系統抵禦已知金鑰攻擊、重送攻擊及竊聽攻擊；「功耗分析」分析比較LPDEM與傳統AES的數據加密消耗的功率。第五章是結論與未來展望。

第二章 相關研究與文獻探討

2.1 物聯網安全性及功耗問題

2.1.1 物聯網安全性問題

IoT將網路通訊技術更加細微的深入人類生活,但同時資訊安全與隱私保護問題也隨之而來[29][30]。資安公司Trustwave SpiderLabs副總裁Lawrence Munro發佈了IoT網路安全的調查報告[31],他表示:「隨著物聯網持續發展,其應用不斷的增加與普及,製造商們急於將產品推出,而忽略了最基礎的網路安全重要性。我們看到許多由於不熟悉資安領域,導致產品存在漏洞的案例,由於更新物聯網產品在本質上極具挑戰性,因此即使發佈補救程式,有些仍然容易受到攻擊,而且多數補救程式還尚未開發出來。廠商需要在網路中測試每個連接到網路上的產品,或是模擬成千上萬種易於被網路攻擊者用來犯罪的潛在新手法。」。他更補充說:「任何使用IP地址的裝置或感測器一旦連到主要控制器的網路,便可能開啟後門,造成毀滅性的資安事件。」[32]。大量的設備互連,產生了可擴展性的問題,2018年八月,台積電傳出生產機台感染電腦病毒,擴散後導致部分生產機台與設備當機,三大廠區停擺多達三天,為目前台灣科技史上最大規模資安事故。

許多研究提出了物聯網安全性方法[19] [33] – [38],Ning與Liu[33]針對資訊傳輸安全、實體機構安全及管理安全等,三個物聯網安全觀點介紹了基於虛實整合的安全性架構。他們使用此架構為單元物聯網與企業物聯網定義資訊安全模型。Li等人[34]認為除了以上提到的三個觀點,也應考慮資料的機密性、完整性、可行性及安全性,作者更指出物聯網是一個混雜異構的網路,需要多面性的安全性方案,包含信任性、算法、身份驗證、權限控制與架構管理。基於這些條件,Horton等人[35]開發並增強私人雲端伺服器及連接機器人之物聯網。Granjal 等人[36]分析現有IoT標準化通訊協定(PHY、MAC、網路及應用)及適用於跨層應用機制的解決方法。Sicari等人[37]提出了物聯網安全領域的挑戰和當前解決方案,其資訊安全問題側重於八個類別:1. 認證問題、2. 存取控制(access control)、3. 機密性問題、4. 隱私、5.

信任制度、6. secure middleware、7. 移動性安全問題及8. 政策執行問題。文中也提出了一些未解決與未來研究的方向。Riahi[38]利用三角金字塔來表示物聯網安全，其中金字塔的四個點分別為人、技術生態系統、程序及智能物件，而四個面用來區分四點相互的作用。因此，還有很多可能的研究可在未來被討論。

2.1.2 物聯網功耗問題

在物聯網的建設裡，能源消耗是一個重要的課題，物聯網的能耗主要來自數據通訊和數據處理，其包括傳輸數據量，數據編碼，數位類比轉換等。因此，當需要處理物聯網資訊安全時，能耗會更加嚴重。Heer 等人[39]表明，為了平衡網路性能和能耗，複雜的加密方法不應用於物聯網。Trappe等人 [23]指出，由於物聯網中終端設備的能源和存儲空間有限，不適合計算複雜的密碼學，傳統常規的密碼學並不適合物聯網系統，他們建議重複使用現有方程式，例如，使用實體層資訊來檢查發射器和接收器的位置。

為了延長物聯網設備的使用壽命，輕量化運算過程以避免過多的能源損耗，為必然的選項，許多研究[43] - [45]都提出了關於物聯網能源管理的方案。Kotamsetty和Govindarasu [40]提出了一種減少對加密數據執行查詢處理時物聯網延遲的方法，能在取得剩餘的加密訊息的同時對另一組數據執行計算工作。Salami 等人 [41]利用認證密鑰加密智能家居的資料，強調加密過程很簡單，不需要複雜的認證。Bui 等人 [42]利用簡單的排列及移位用於密鑰/數據存儲的暫存器來提供低功耗AES架構，以減小電路尺寸和功耗，並提出了一種名為clock gating用於S-box上的省電的低功耗技術。Shafagh等人[43]提出了一種用於物聯網的加密算法。該方法允許在雲端安全地儲存加密的數據，並能即時的對加密數據作查詢處理。他們利用替代的輕量級加密算法以適應物聯網設備的計算限制。該系統使用end-to-end系統取代原Web應用程式通訊，將來自設備的加密數據儲存在雲端上，在客戶端執行數據的加密/解密。

安全性是當前在許多應用中的相當關注的議題，例如雲端的服務、智能醫療保健等[45] - [47]。然而，在物聯網上採用複雜的安全（或加密）方法會為物聯網設備消耗大量功率/能源，更糟糕的是，可能會降低網絡效能[48]。因此這是安全性，效

能，電路面積，網絡負載量和功耗/能耗之間的權衡[49]，在本論文中，提出 LPDEM 為物聯網數據加密提供安全但低功耗的方法。

2.2 LoRaWAN 性能與安全性

De Silva 等人在研究中[50]介紹了 LoRaWAN 架構及協定，並比較了 LoRaWAN、Sigfox、NB-IoT、LET-M 的電池壽命、數據傳輸數率、通訊範圍及安全性等性能表現，作者更進一步地指出在廣域通訊上，LoRaWAN 的功率損耗表現優於其他 LPWAN 技術。然而，隨著範圍的增廣，更多的閘道意味著需要提升其網路表現。Bankov 等人[51]利用數學方法分析模擬 LaRaWAN 發展的限制性，根據 LoRaWAN 的規範，當終端設備的數量大幅成長時，單個閘道傳輸的總數據量會迅速增加，容易導致資料錯誤率上升。其中一個解決方案便是在 LoRaWAN 中增置更多閘道，這將導致網路成本增加。Mikhaylov 等人[52]也指出了雷同的問題，儘管 LoRaWAN 在低傳輸量之下具較高的節點涵蓋率與擴充性，但在高傳輸量之下，其可靠度、傳輸延遲及網路效能則大受影響。由上述文獻探討可以得知 LoRaWAN 較適用於低傳輸量的物聯網環境中，若使用於高傳輸量的環境，則須利用多 gateway 的佈建改善其整體效能。

其他一些研究側重於LoRaWAN的安全問題[53] - [55]。Miller [53]分析了可能對LoRaWAN的攻擊，並主張增強加密密鑰生成程序和密鑰管理原則。在LoRaWAN中，所有終端設備，閘道和服務器都應該有自己的用戶驗證和密鑰保護原則，以確保通訊的安全性。Tomasin等人 [54]與Aras 等人[55]再研究裡調查了LoRaWAN的安全漏洞。Tomasin等人聲稱在新添加終端設備時可能會發生重送攻擊和DoS攻擊。除了重送攻擊，Aras等人進一步表明，廣域通訊可能遭受無線電干擾和蟲洞攻擊。

Naoui等人 [56]提出了一種新的LoRaWAN安全性結構，其使用代理節點來執行閘道的部分功能。這些代理節點評估相鄰代理節點的可靠性後，創建可靠度表然後將其傳遞給所有終端設備。根據該表，每個終端設備選擇具有最高可靠性並且可用的代理節點來傳送數據。Girard [57]利用受信任的第三方來保護兩個會話密鑰的生成過程。Kim和Song[58]認為會話密鑰生成和更新的過程仍存在一些安全問題。他們

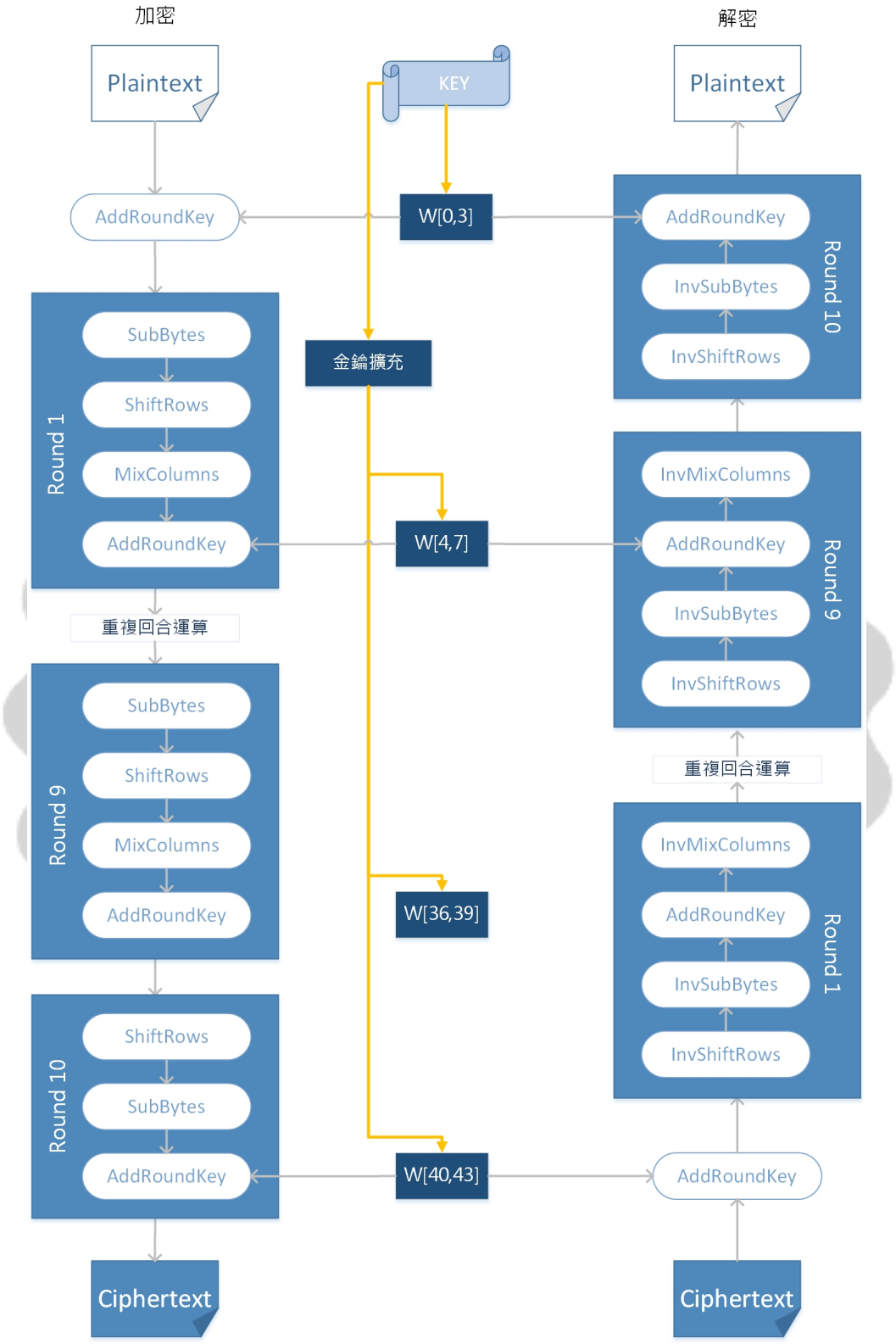
使用名為NwkKey的新網路密鑰來保護兩個會話密鑰的更新過程，而不使用受信任的第三方。他們的實驗結果證明此舉可以提高安全性水平；然而，密鑰生成和密鑰更新處理時間也增加，比原始系統消耗更多能量。

在[56] - [58]中，可以看出，為了提高LoRaWAN的安全等級，通常會採取一些複雜的操作或程序，但是此舉導致數據加密、數據解密和資料驗證過程消耗的功率增加。McGrew[59]觀察到許多加密算法（包括AES）過於複雜，無法降低物聯網設備的功耗，他提出了一種需要經過身份驗證的加密方法，稱為Authenticated Encryption with Replay protection（以下簡稱AERO）。AERO需驗證明文和序列編號。變量序列編號全部或部分的數字隱藏在傳遞的訊息中。因此，攻擊者無法透過收集大量消息來獲取加密密鑰。由於AERO的加密方法操作簡單，因此可以降低功耗。但是，AERO的安全級別仍需要確認。

2.3 AES-128 加密法

AES 加密法（Advanced Encryption Standard，AES) [18]是一種對稱區塊的加密方法，每一區塊具有固定的大小。AES加密法取代了DES加密法，為目前公認安全性較高的加密機制之一，其安全快速的加解密機制主要是透過強韌的代數運算及多回合加密建構而成。其設計具高度彈性，可以在軟體、硬體及韌體上實作，並利用查表或是使用完整定義的算術結構去改變加密的過程。

其金鑰長度可以是128,192和256位元，分別由10，12和14個重複的加密（也稱為回合）組成。圖四為AES的加密十次的結構圖，由圖可知，AES在加密、解密的過程中，在未進入回合運算前，都會先執行AddRoundKey一次，之後每輪再執行一次。每輪都包括四個處理步驟，SubBytes、ShiftRows、MixColumns和AddRoundKey。在加密的過程中，會做金鑰的擴充，將數據與由加密金鑰生成的回合金鑰混合。由於AES是屬於對稱金鑰加密，所以左邊的加密跟右邊的解密系統是一樣的架構，只是在解密時是進行加密過程的反向運算，一樣經過十個回合之後，便可以得到跟原本相同的明文。



圖四 AES加密系統結構圖[60]

AES加密法的四個步驟如下說明：

1. SubBytes—一種可逆和非線性變換，採用16個相同的256位元尋找表（例如S-box)，用於將數據塊的位元組替換成相對應的位元組。S-box與Galois Field GF（2^8）中的乘法反元素有關，其具有良好的非線性特性。

2. ShiftRows—將十六個字節均分成四組，以字節為單位進行亂序處理，根據定義的偏移量循環位移數據塊的行來執行字節元位移，例如：第二、第三和第四行向左移分別一個、兩個和三個字元。

3. MixColumns—將數據塊的每一列與 GF（2^8）中的多項式相乘。SubBytes 和 MixColumns 也可以組合成大型尋找表 Look-Up-Table (LUT)，而不是單獨計算，MixColumns 接受四個位元組的輸入，運算出另外四個位元組，每一個輸入的位元組都會對輸出造成影響。因此 ShiftRows 和 MixColumns 兩步驟可以為密碼系統提供擴散性。

4. AddRoundKey—在此步驟，由原始的密鑰產生的回合金鑰會與數據塊混合，原矩陣的每一個位元都會與回合金鑰做 XOR 運算

在 AES 裡，透過金鑰擴展（key-expansion），產生出每一回合中所使用的回合金鑰程序。假設回合數為N_r，輸入一個長度為 128 位元的加密金鑰，經過擴展程序後便產生$N_r + 1$個長度為 128 位元的回合金鑰。金鑰擴展程序在計算字組時，可以選擇使用查表或者在伽羅瓦體 GF（2^8）下計算最左邊位元組的值。AES 中每個回合金鑰都是由上一個回合金鑰算出來的。然而，因為計算中使用 SubWord 轉換程序，所以回合金鑰之間的關係是非線性的。加入回合常數的步驟也確保了每個回合金鑰都不會和上一個相同。

Huang 等人[61]提出的動態累積移位替換方法(The dynamic accumulated shifting substitution Algorithm，DASS)是一種單向不可逆將明文加密成密文的方程式。然而在 DASS 中，位移計數器（shifting counter，ct）除了是線性變化，在查詢 S-Box 時也僅有 0 到 8 的範圍，這將導致無法防禦暴力攻擊法（Brute-force Attacks)。為了解

決此問題，Liu 等人在研究[62]內提出的 Generating a Dynamic Box by using Input String (以下簡稱 GDBIS)，其中 The Enhanced DASS Algorithm（以下簡稱 EDASS），同為單向不可逆地將明文加密成密文，EDASS 輸入 128 位元的明文與 16*16 的隨機矩陣，輸出 128 位元的密文。其非線性的增加動態位移次數(dynamic shifting count，*dsc*）與查詢隨機矩陣，使得輸出的密文擁有極高的機密性。換句話說，只要輸入的明文有些微變化，輸出都會顯著的不同。細部的說明提供於 3.1.1 章節。

在討論如何改善AES安全性的文獻中，許多研究討論區塊加密[63]與如何提升S-Box複雜性[64] –[66]。在AES加密過程中，SubBytes透過查表來加解密數據，快速且高敏感性的加密輸出，輸入微異的明文，其輸出之密文具高度的差異。S-Box在加密過程中使用到了兩次：一次是在密鑰(Cipher Key)擴展生成輪密鑰(Round Key)的時候，另一次則是在輪加密步驟中的替代位元運算（SubBytes）。

然而AES裡的查表內容(例如S-Box)多為固定，唯一非線性的為其查表的過程，固定內容大幅地降低了加密方法的安全等級。Liu等人研究[62]裡介紹的D-Box生成算法（the D-Box generation，DBG），利用使用者定期更新輸入設定與加密金鑰去計算出相關的動態矩陣（dynamic box，D-Box），D-Box為每k天更新的查詢表。本文參考GDBIS算法，將S-Box置換成D-Box，其降低原S-Box線性、固定的缺點，不可逆、一樣具高明感性的加密輸出，且對於不同輸入設定產生之查表衝突性低，顯著的提升AES加密的安全性，其完整的算法會在3.1.2章說明。

第三章 低功耗資料加密方法

為了建立安全且低功耗的通訊，本論文提出低功耗資料加密方法(Low Power Data Encryption Method，LPDEM)，分成兩部分，分別是AES加密改善與資料加密方法。AES加密法改善的部分，透過The Enhanced DASS Algorithm[62]，輸入128位元的明文與16*16的隨機矩陣，單向不可逆的加密輸出128位元的密文，接著產生相關的動態的dynamic box (D-Box)，取代掉原有的S-Box。在資料加密方法的部分，將用於加密應用層負載的AppSKeys與D-Box，每k天更新一次，更新方法在章節3.2做說明。

LPDEM定期更新用於密鑰擴展之輸入，提高攻擊難度，減少加密循環次數，達到安全且降低功耗之目的。下文先介紹單向密鑰擴展與D-Box的生成，最後說明AppSKeys$_{\alpha\text{-}new}$與D-Box$_{\alpha\text{-}new}$的更新程序。

3.1 AES 加密改善

在AES加密法之SubBytes步驟中，透過查詢S-Box加密及解密資料，S-Box通常用於模糊金鑰與密文之間的關聯性，為了增強AES強度， S-Box由動態的dynamic box (D-Box) 取代。由D-Box生成算式（Dynamic Box Generator algorithm，以下簡稱DBG）生成，輸入字串計算出16×16的D-Box矩陣，其內容為隨機從00到FF不重複的元素，導自三個內部密鑰（Dynamic Key，包含DK_1、DK_2及DK_3）與三個內插陣列（Insert Array，包含$IA1$、$IA2$及$IA3$），在安全性上得到了卓著改善。

以下先介紹相關參數與計算的符號說明，如表一所示，接著一一說明算式步驟。

表一 參數說明

參數名稱	參數說明
P	明文，n bits，8 bits 為一區塊， $P = p_1 p_2 \dots p_k,\ k = \mathrm{n}/8$。
C	密文，n bits，8 bits 為一區塊， $C = c_1 c_2 \dots c_k,\ k = \mathrm{n}/8$。
dsc	Dynamic shifting count，一開始等 256，隨著計算過程，會產生新的非線性 *dsc*。
ISKs	Initial Serial Keys，透過算式 EDASS_expansion(*P*) 與算式 Binaryadder_expansion (*P*)得出，包含ISK_1與$\bar{ISK}_2$，長度為 1028 bits。
DKs	Dynamic Keys，源於 *ISKs*，透過算式 EDASS 得出，包含DK_1、DK_2及DK_3，長度為 1028 bits。
FA	Flag Array，用來辨識 D-Box 中的數值，確保 D-Box 裡每個數值的獨特性，如果其索引內容為"T"(或"F")，表示該索引已經(未)被使用，使其內容不會產生衝突。
IAR	Residual Insert Array，組成為 *FA* 內剩餘未被選中元素。
RLR	Real length of *IAR*，為 *IAR* 的實際長度。
*IA*1	第一個內差矩陣，內容源自DK_1，透過 FA 的幫助去除剩餘的元素。
*IA*2	第二個內差矩陣，內容源自DK_2與DK_3，透過 FA 的幫助去除剩餘的元素。
*IA*3	第三個內差矩陣，其內容為重整的 *IAR*。
RLi	Real length，包含RL_1、RL_2及RL_3，為內差矩陣 *IA1*、*IA2*、*IA3* 的實際長度。

計算的符號說明：

假設p_i為明文 P 的區塊，c_i為密文 C 的區塊，k 為加密的金鑰

$\otimes$：Exclusive-OR，輸入為異時，輸出為 True。

$\odot$：Exclusive-NOR，輸入為同時，輸出為 True。

$+_2/-_2$：二進制的加減法。

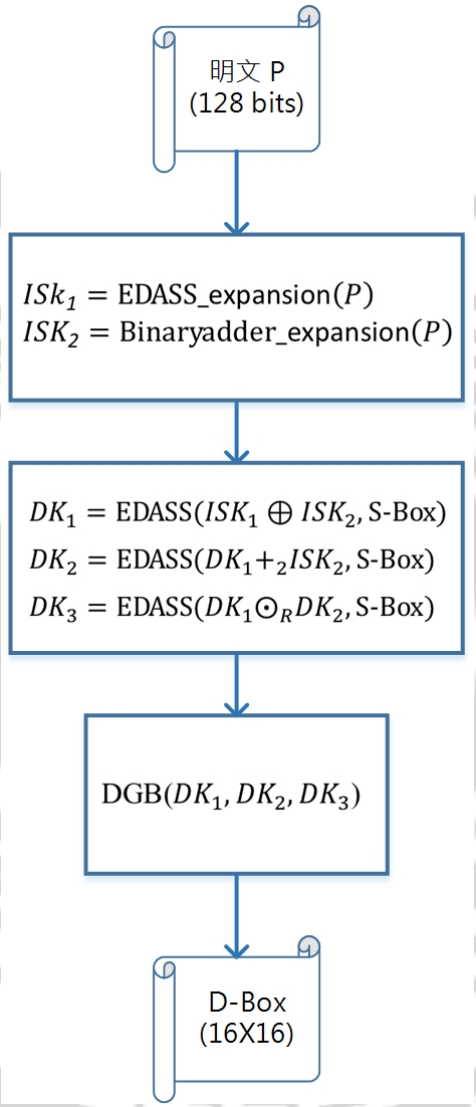
$\odot_R$：Rotate-Equivalence operator，

加密時：$c_i = p_i \odot_R k = p_{iR} \odot k$，

其中p_{iR}為明文p_i連續的順時鐘旋轉 h bits，$h = |k|/4$。例如$|k| = 128$，則p_i將旋轉 32 bits。

解密時：$p_i = c_i \odot_{IR} k$ ＝逆時鐘的旋轉$(c_i \odot k)|k|/4$ bits

以下分成三個步驟，在算式一EDASS中，將輸入與隨機矩陣加密成密文；透過算式二與算式三將輸入擴展至1028 bits，產生*ISK*s及*DKs*；最後由算式四總合前面，產生D-Box。以下為D-Box生成的流程圖，詳細算式隨後說明。



圖五 D-Box生成流程圖

Algorithm 1：EDASS(*P*, R-Box)

輸入：一明文P、一16×16的random-box(以下簡稱R-Box)。

輸出：密文*C*。

1. Let $P = p_1 p_2 \ldots p_k$, and let $C = c_1 c_2 \ldots c_k$, where $k = \mathrm{n}/8$; /* $|p_i|=|c_i|=8$ bits, $1 \le i \le k$ */
2. *dsc* = 256; /**dsc*: dynamic shifting count*/
3. For *i*= 1 to *k* {
4. *vp*[*i*] = Int(p_i); /* Int(p_i) retrieves the ACSII code of p_i*/
5. *dsc* = *dsc* + (*vp*[*i*] +1)**i*;} /* yielding non-linear increment of *dsc* */
6. *vp*[0] = *vp*[1] + *vp*[*k*];
7. For *i*= 1 to *k* {
8. *dsc* = (*dsc* + (*vp*[*i*-1])**i*) mod 65536;
9. *ch* = str((*vp*[*i*]+ *dsc*) mod 256); /* str(*X*) returns ASCII code of *X* */
10. c_i = the ciphertext which is the corresponding content in the R-Box after *ch* is looking up by using R-Box;}

算式說明：

1. 將明文P及密文C每8 bits為一區塊，如式3-1所示。

$$\begin{cases} P = p_1 p_2 \ldots p_k, \ k = \frac{n}{8} \\ C = c_1 c_2 \ldots c_k, \ k = \frac{n}{8} \end{cases} \quad \text{.....(式3-1)}$$

2. *dsc*一開始設定為256。

3. 透過*Int*(p_i)輸入明文*P*區段p_i轉成十進制整數輸出，存至矩陣$vp[i]$中，如式3-2中所示。

$$\begin{cases} vp[i] = Int(p_i) \\ dsc' = dsc + (vp[i]+1)\times i' \end{cases}, \ 1 \le i \le \text{k} \quad \text{.....(式3-2)}$$

4. 由於上式$\text{i} = 1 \text{ to k}$，$vp[0]$為空值，因此$vp[0]$由$vp[1]$與$vp[k]$相加得出，如式3-3所示。

$$vp[0] = vp[1] + vp[k] \quad \text{.....(式3-3)}$$

5. 透過str(X)輸入十進制整數轉換成二進制輸出，存放在ch，如式3-4中所示。

$$\begin{cases} dsc' = (dsc + (vp[i-1])\times i) \bmod 65536 \\ ch = \text{str}\big((vp[i] + dsc') \bmod 256\big) \end{cases}, 1 \le i \le \text{k} \quad \text{.....(式3-4)}$$

6. 接著利用ch查詢R-Box，最終得出密文區塊c_i。

在生成 Initial Serial Keys 的步驟中，需要透過算法二與算法三擴展至 1024 bits，如下所示。

Algorithm 2： EDASS_expansion (*P*)

輸入：一明文 *P*。

輸出：1024 bits 的 *P''*。

1. Do {
2. *P*'= EDASS (*P*, R-Box);
3. *P* = *P*||*P*';}
4. while (|*P*|<1024)
5. *P* = EDASS (*P*, R-Box);
6. return Right(*P*, 1024); /* return the 1024 right most bits of *P* */

算式說明：

1. 透過EDASS將明文P加密得到新的P'，與原明文串接，重複動作直到擴充成1024 bit，如式3-5所示。

$$\text{Do } \begin{cases} P' = \text{EDASS}(P, \text{R-Box}) \\ P'' = P \parallel P' \end{cases}, \text{while } |P| < 1024 \quad \text{(式3-5)}$$

Algorithm 3：Binaryadder_expansion (P)

輸入：一明文P。

輸出：1024 bits的P'。

1. While (|P|<1024){
2. $P = (P||P)+2(P||P)$; }
3. return Right(P,1024);

算式說明：

1. 將四個輸入的明文P兩兩串接後相加得到新的P'，重複動作直到擴充成1024 bit，如式3-6所示。

$$\text{Do } \{P' = (P \parallel P) +_2 (P \parallel P)\}, \text{while } \{|P| < 1024\} \quad \text{(式3-6)}$$

生成 Initial Serial Keys（*ISK*s），如式 3-7 所示。

$$\begin{cases} ISK_1 = \text{EDASS\_expansion}\ (P) \\ ISK_2 = \text{Binaryadder\_expansion}(P) \end{cases} \quad \text{.....(式3-7)}$$

產生動態金鑰（Dynamic keys，*DK*），如式3-8所示。

$$\begin{cases} DK_1 = \text{EDASS}(ISK_1 \oplus ISK_2, \text{R-Box}) \\ DK_2 = \text{EDASS}(DK_1 +_2 ISK_2, \text{R-Box}) \\ DK_3 = \text{EDASS}(DK_1 \odot_R DK_2, \text{R-Box}) \end{cases} \quad \text{.....(式3-8)}$$

Algorithm 4：DGB(DK_1, DK_2, DK_3)

輸入：DK_1、DK_2、DK_3

輸出：D-Box

1. { Let $DK_1 = KA_0 || KA_1 || ... || KA_{127}$, where KA_j is 8 bits in length for all j, $0 \leq j \leq 127$;
2. Let $DK_2 = KB_0 || KB_1 || ... || KB_{127}$, where KB_j is 8 bits in length for all j, $0 \leq j \leq 127$;
3. Let $DK_3 = KC_0 || KC_1 || ... || KC_{127}$, where KC_j is 8 bits in length for all j, $0 \leq j \leq 127$;
4. For j = 0 to 127
5. {*KA*[*j*] = Int(KA_j); *KB*[*j*] = Int(KB_j); *KC*[*j*] = Int(KC_j); *IA*1[*j*]=0; }
6. For j = 0 to 255
7. {D-Box[*j*]=0; *IA*2[*j*]=0; *IAR*[*j*]=0; *IA*3[*j*]=0;}
8. *RL*1=0; *RL2*=0; *RLR*=0;
9. For i = 0 to 127
10. If (*FA*[*KA*[*i*]] = "F")

11. {*IA*1[*RL1*] = *KA*[*i*]; *FA*[*KA*[*i*]] = "T"; *RL*1 = *RL*1+1;}

12. For *i* = 0 to 127

13. {If (*FA*[*KB*[*i*]] = "F")

14. {*IA*2[*RL*2] = *KB*[*i*]; *FA*[*KB*[*i*]] ="T"; *RL*2 = *RL*2+1;}

15. If (*FA*[*KC*[*i*]] = "F")

16. {*IA*2[*RL*2] = *KC*[*i*]; *FA*[*KC*[*i*]] = "T"; *RL*2=*RL*2+1;}}

17. For *i* = 0 to 255

18. If (*FA*[*i*] = "F")

19. {*IAR*[*RLR*] = *i*; *RLR* = *RLR*+1;}

20. *HL* = *RLR*/2;

21. For *i* =0 to *HL*-1

22. {*IA*3[*i*] = *IAR*[*HL*+*i*]; *IA*3[*HL*+*i*] = *IAR*[*i*];}

23. If (*RLR* is odd)

24. *IA*3[*RLR*-1] = *IAR*[*RLR*-1];

25. *t*1=0; *t*2=0; *t*3=0; *j*=0;

26. while ($j \leq 255$)

27. If (*t*1 < *RL*1){

28. {D-Box[*j*] = *IA*1[*t*1]; *j* = *j*+1; *t*1 = *t*1+1;}

29. If (*t*2 < *RL*2)

30. {D-Box[*j*] = *IA*2[*t*1]; *j* = *j*+1; *t*2 = *t*2+1;}

31. If (*t*3 < *RLR*)

32. {D-Box[*j*]=*IA*3[*t*3]; *j* = *j*+1; *t*3 = *t*3+1;}}}

算式說明：

1. 由式3-8算出的動態金鑰DK_1, DK_2, DK_3，每8 bit為一區塊，如式3-9所示。

$$\begin{cases} DK_1 = KA_0 \parallel KA_1 \parallel \cdots \parallel KA_{127}, \text{where } KA_j \text{ is 8 bits in length for all } j, 0 \le j \le 127 \\ DK_2 = KB_0 \parallel KB_1 \parallel \cdots \parallel KB_{127}, \text{where } KB_j \text{ is 8 bits in length for all } j, 0 \le j \le 127 \\ DK_3 = KC_0 \parallel KC_1 \parallel \cdots \parallel KC_{127}, \text{where } KC_j \text{ is 8 bits in length for all } j, 0 \le j \le 127 \end{cases}$$

.....(式3-9)

2. 初始化所有參數，如式3-10至3-12所示。

$$IA1[k] = 0, 0 \le k \le 127 \text{.....(式3-10)}$$

$$\begin{cases} \text{D-Box}[k] = 0 \\ IAR[k] = 0 \\ IA2[k] = 0 \\ IA3[k] = 0 \\ FA[k] = "F" \end{cases}, \ 0 \le k \le 255 \text{.....(式3-11)}$$

$$RL1 = 0, RL2 = 0, RLR = 0 \text{.....(式3-12)}$$

3. 透過$Int(p_i)$輸入$KA_j.KB_j.KC_j$轉成十進制整數輸出，存至$KA[j].KB[j].KC[j]$，如式3-13所示。

$$\begin{cases} KA[j] = \text{Int}(KA_j) \\ KB[j] = \text{Int}(KB_j) \\ KC[j] = \text{Int}(KC_j) \end{cases}, 0 \le j \le 127 \text{.....(式3-13)}$$

4. 透過標誌矩陣（$FA[i]$）確保元素$KA[j].KB[j].KC[j]$未使用。$IA1$由$KA[i]$檢索而成，如式3-14所示；$IA2$由$KB[i]$與$KC[i]$檢索而成，如式3-15所示。

$$If\ (FA[KA[i]] = "F"), \begin{cases} IA1[RL1] = KA[i] \\ FA[KA[i]] = "T" \\ RL1 = RL1 + 1 \end{cases}, 0 \le i \le 127 \text{.....(式3-14)}$$

$$\begin{cases} If\ \left(FA\left[KB[i]\right] = \text{"F"}\right), \begin{cases} IA2[RL2]\ =\ KB[i] \\ FA[KB[i]]\ =\ \text{"T"} \\ RL2\ =\ RL2+1 \end{cases} \\ If\ \left(FA\left[KC[i]\right] = \text{"F"}\right), \begin{cases} IA2[RL2]\ =\ KC[i] \\ FA[KC[i]]\ =\ \text{"T"} \\ RL2\ =\ RL2+1 \end{cases} \end{cases}, 0 \le i \le 127 \quad \text{(式3-15)}$$

5. 檢查$FA[i]$內尚未使用的元素，並將之存在$IAR[i]$，其實際長度為RLR，如式3-16所示。

$$\text{If}\ (FA[k] = \text{"F"}), \begin{cases} IA\text{R}[RLR]\ =\ k \\ RLR = RLR+1 \end{cases}, 0 \le k \le 255 \quad \text{(式3-16)}$$

6. 將$IAR[i]$分成前後兩部分，做前後置換處理後存至$IA3[i]$，如式3-18所示。如果RLR為奇數，則$IAR[RLR\text{-}1]$直接存至$IA3[RLR\text{-}1]$，如式3-19所示。

$$HL = \left\lfloor \frac{RLR}{2} \right\rfloor \quad \text{(式3-17)}$$

$$\begin{cases} IA3[i] = IAR[HL+i] \\ IA3[HL+i] = IAR[i] \end{cases}, 0 \le i \le HL-1 \quad \text{(式3-18)}$$

$$\text{If}(RLR\ \text{is odd}), \{IA3[RLR-1] = IAR[RLR-1]\} \quad \text{(式3-19)}$$

7. 依序將$IA1[i]$、$IA2[i]$、$IA3[i]$值存至D-Box$[i]$，如式3-21所示

$$\text{t1=0, t2=0, t3=0, m=0} \quad \text{(式3-20)}$$

$$\begin{cases} If\ (t1 < RL1), \{\text{D-Box}[m] =\ IA1[t1], m = m+1, t1 = t1+1\} \\ If(t2 < RL2), \{\text{D-Box}[m] =\ IA2[t2], m = m+1, t2 = t2+1\} \\ If(t3 < RLR), \{\text{D-Box}[m] =\ IA3[t3], m = m+1, t3 = t3+1\} \end{cases}, \text{while}\ (m \le 255)$$

$$\quad \text{(式3-21)}$$

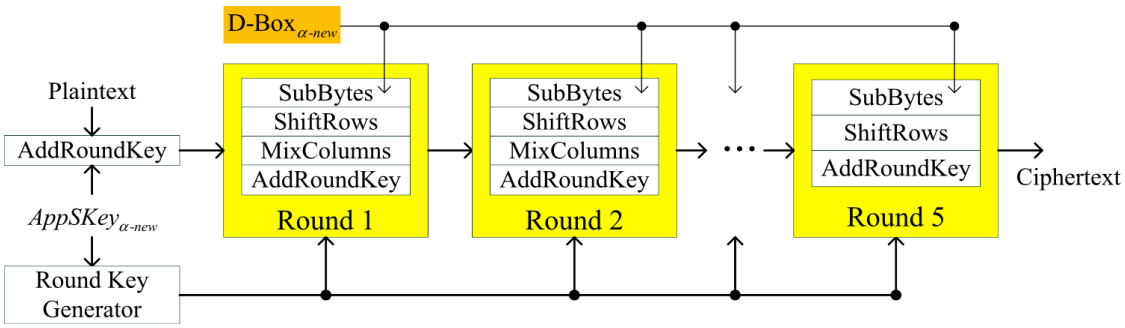
3.2 資料加密方法

3.2.1 Simplified-AES加密法

在章節 2.1 章提到，物聯網的功耗主要來自於數據通訊與數據處理，傳統常規的密碼學並不適合用於物聯網，在物聯網上使用複雜的加密方法會消耗大量能源，違背初衷。如章節 2.2 章介紹的 LoRaWAN 安全規範，LoRaWAN 利用 AppSKey 加密應用層的負載；NwkSKey 產生 MIC 碼給 MAC 層。在章節 2.4 章提及 AES-128 加密法在終端裝置，每輪將數據與由加密金鑰生成的回合金鑰混合，重複十個加密循環。

為了延展物聯網設備的壽命，必須於安全性、效能、功耗等之間取得權衡，而但當 AppSKey 及 D-Box 每 k 天更新一次，對於駭客攻擊加密過程是困難的，為了降低計算的複雜性及降低終端裝置的功耗，基本五個加密循環就已足夠。圖五說明 Simplified-AES 加密法。Simplified-AES 加密法跟傳統的 AES 類似，除了只有五次的循環，每回合都有 SubBytes, ShiftRows, MixColumns, and AddRoundKey 步驟。每回合 $\text{AppSKeys}_{\alpha\text{-}new}$ 都會被置入且產生該回合金鑰；並在 SubBytes 步驟加入 $\text{D-Box}_{\alpha\text{-}new}$。在經過五次的加密循環，即產生密文至 MAC 層的負載。

AppSKey 及 D-Box 更新程序於章節 3.2.2 章做說明。

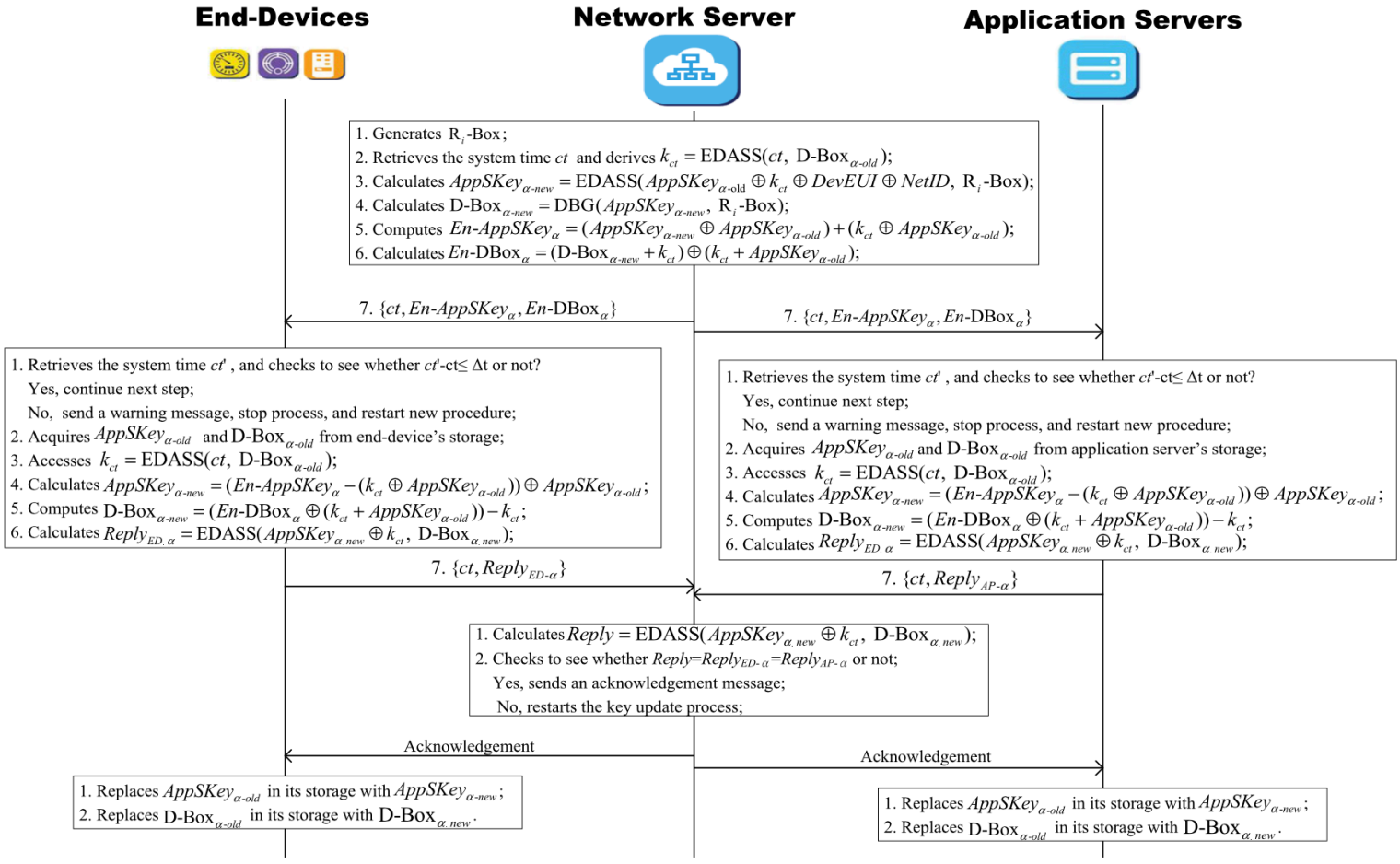


圖六 Simplified-AES加密程序

3.2.2 AppSKey及D-Box更新程序

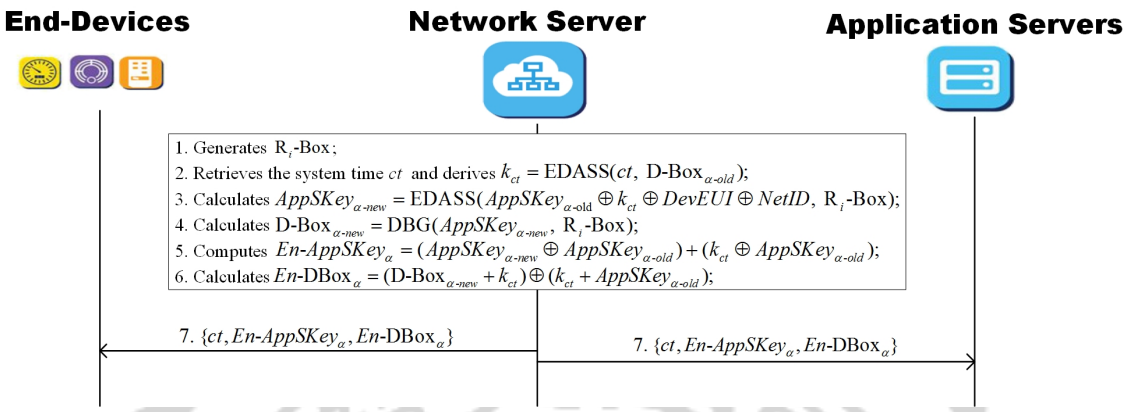
為了降低計算複雜度與提升LoRaWAN的安全性，將那些連至同一通訊閘的終端裝置視為一個群體，它們自身用於加密應用層的動態加密金鑰（AppSKey）及D-Box依照網路管理者的定義，每k天更新一次，基於安全性考量，不同的群體會有不同的k。終端裝置α最新的AppSKeys與D-Box表示成AppSKeys$_{\alpha\text{-}new}$及D-Box$_{\alpha\text{-}new}$。

當執行更新程序時，網路服務器產生新的AppSKey與D-Box送至該群體的所有終端裝置與其應用服務器，更新程序以序列圖表示由圖四說明。下文說明會拆解成四部分於章節3.2.3章至章節3.2.6章做說明。



圖七 AppSKey與D-Box更新流程序列圖

3.2.3 第一階段



圖八 AppSKey與D-Box更新流程序列圖-1

在更新程序之初，由網路服務器根據暫存器內的資料計算出*Time key*(k_{ct})、$AppSKey_{\alpha-new}$、DBox$_{\alpha-new}$，並進一步加入k_{ct}加密計算出$En\text{-}AppSKey_{\alpha}$、$En$-DBox$_{\alpha}$，最後傳送$\{ct, En\text{-}AppSKey_{\alpha}, En\text{-DBox}_{\alpha}\}$給終端裝置end-device$_{\alpha}$及其應用服務器。網路服務器的詳細動作如下：

1. 產生一個亂數表R_i-Box給連接至通訊閘G_i的終端裝置end- device$_{\alpha}$ (α =1,2,…,m)；

2. 回報system time (ct)並獲得一個128位元的time key (k_{ct})，

   k_{ct}=EDASS(ct,D-Box$_{\alpha\text{-}old}$)；

3. 計算$AppSKeys_{\alpha\text{-}new}$=EDASS($AppSKey_{\alpha\text{-}old} \otimes k_{ct} \otimes DevEUI \otimes NetID, R_i$-Box)，其中$DevEUI$為符合IEEE EUI-64標準的地址格式，$NetID$是24位元的認證碼；其中最後的五個有效字節（5 LSBs）為$NetID$用來區分重複LoRa網路地址，其他部分則由網路服務器作定義；

4. 計算D-Box$_{\alpha\text{-}new}$=DBG($AppSKeys_{\alpha\text{-}new}, R_i$-Box)；

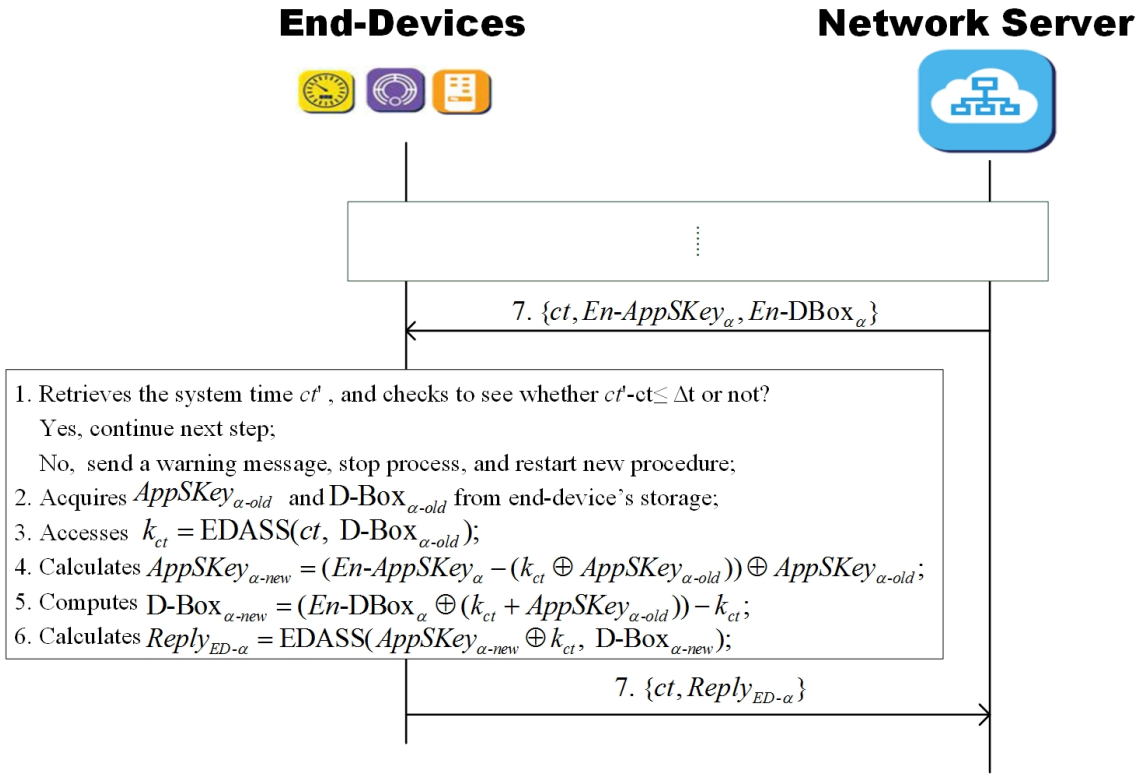
5. 計　　算　　$En\text{-}AppSKey_{\alpha} = (AppSKeys_{\alpha\text{-}new} \oplus AppSKeys_{\alpha\text{-}old}) + (k_{ct} \oplus AppSKeys_{\alpha\text{-}old})$；

6. 計算$En\text{-DBox}_{\alpha} = (\text{D-Box}_{\alpha\text{-}new} + k_{ct}) \oplus (k_{ct} + AppSKeys_{\alpha\text{-}old})$；

7. 傳送$\{ct, En\text{-}AppSKey_{\alpha}, En\text{-DBox}_{\alpha}\}$給終端裝置$\text{end-device}_{\alpha}$及其應用服務器，其中$\alpha$=1,2,…,m。

3.2.4 第二階段

1. 終端裝置面：

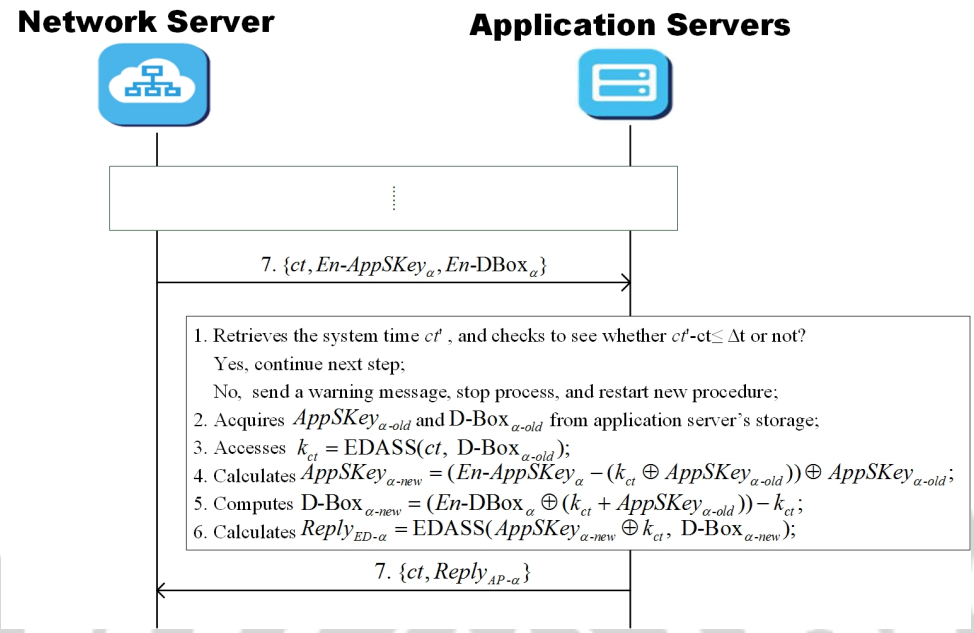


圖九 AppSKey與D-Box更新流程序列圖-2

在end-device$_{\alpha}$接收到更新的訊息後，首先會確認$ct' - ct \leq \Delta t$是否為真，並獲取暫存器內的資料進而計算出*Time key*(k_{ct})、$AppSKey_{\alpha-new}$、$\text{DBox}_{\alpha-new}$。接著計算$\text{Reply}_{\text{ED-}\alpha}$ 並回傳給網路服務器。終端裝置的詳細動作如下：

1. 回報system time (ct')，並確認$ct' - ct \leq \Delta t$是否為真，其中Δt由最大傳輸延遲及金鑰更新程序時間預先定義。如果為否，將傳送警告訊息給網路服務器，停止當前金鑰更新程序，通知網路服務器進行新的更新；
2. 從內部的儲存器獲取$AppSKey_{\alpha\text{-}old}$及$\text{D-Box}_{\alpha\text{-}old}$；
3. 利用更新訊息中的ct存取(accesses)$\text{D-Box}_{\alpha\text{-}old}$及時間金鑰$k_{ct}$；
4. 計算 $AppSKeys_{\alpha\text{-}new} = \left(En\text{-}AppSKey_{\alpha} - (k_{ct} \oplus AppSKeys_{\alpha\text{-}old})\right) \oplus AppSKeys_{\alpha\text{-}old}$；
5. 計算$\text{D-Box}_{\alpha\text{-}new} = En\text{-}\text{DBox}_{\alpha} \oplus (k_{ct} + AppSKeys_{\alpha\text{-}old})$；
6. 計算$Reply_{ED\text{-}\alpha} = \text{EDASS}(AppSKeys_{\alpha\text{-}new} \oplus k_{ct}, \text{D-Box}_{\alpha\text{-}new})$；
7. 傳送$\{ct, Reply_{ED-\alpha}\}$給網路服務器。

2. 應用服務器面：



圖十 AppSKey與D-Box更新流程序列圖-3

在應用服務器接收到更新的訊息後，首先會確認$ct' - ct \leq \Delta t$是否為真，並獲取暫存器內的資料進而計算出$Time\ key(k_{ct})$、$AppSKey_{\alpha-new}$、$\text{DBox}_{\alpha-new}$。接著計算$\text{Reply}_{\text{AP-}\alpha}$ 並回傳給網路服務器。應用服務器的動作如下：

1. 回報system time (ct')，並確認$ct' - ct \leq \Delta t$是否為真，其中Δt由最大傳輸延遲及金鑰更新程序時間預先定義。如果為否，將傳送警告訊息給網路服務器，停止當前金鑰更新程序，通知網路服務器進行新的更新；

2. 從內部的儲存器獲取$AppSKey_{\alpha\text{-}old}$及$\text{D-Box}_{\alpha\text{-}old}$；

3. 利用更新訊息中的ct存取(accesses)$\text{D-Box}_{\alpha\text{-}old}$及時間金鑰$k_{ct}$；

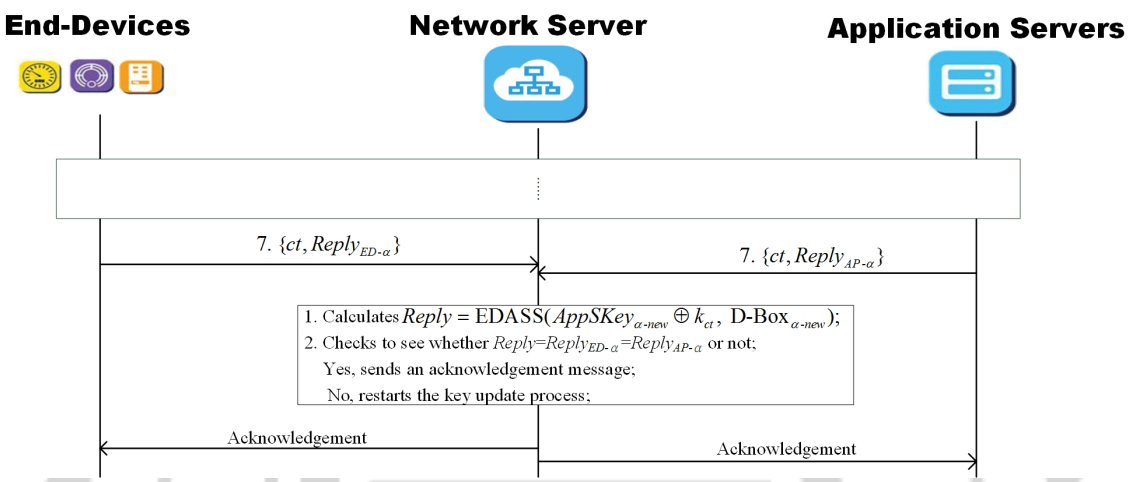
4. 計算 $AppSKeys_{\alpha\text{-}new} = \left(En\text{-}AppSKey_{\alpha} - (k_{ct} \oplus AppSKeys_{\alpha-old})\right) \oplus AppSKeys_{\alpha\text{-}old}$；

5. 計算$\text{D-Box}_{\alpha\text{-}new} = En\text{-}\text{DBox}_{\alpha} \oplus (k_{ct} + AppSKeys_{\alpha\text{-}old})$；

6. 計算$Reply_{ED\text{-}\alpha}$ = EDASS($AppSKeys_{\alpha\text{-}new} \oplus k_{ct}$, D-Box$_{\alpha\text{-}new}$)；

7. 傳送$\{ct, Reply_{AP\text{-}\alpha}\}$給應用服務器。

3.2.5 第三階段



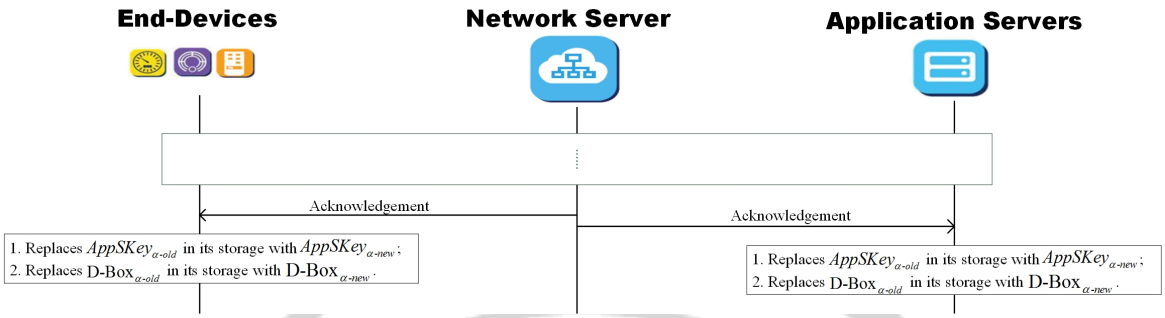
圖十一 AppSKey與D-Box更新流程序列圖-4

當接收到分別來自end-device$_\alpha$及應用服務器的回應訊息後，首先計算$Reply$，接著檢查$Reply = Reply_{ED\text{-}\alpha} = Reply_{AP\text{-}\alpha}$是否為真。網路服務器的詳細動作如下：

1. 計算$Reply$ = EDASS($AppSKeys_{\alpha\text{-}new} \oplus k_{ct}$, D-Box$_{\alpha\text{-}new}$)；

2. 檢查$Reply = Reply_{ED\text{-}\alpha} = Reply_{AP\text{-}\alpha}$是否為真；如果為真，傳送確認訊息給每一個end-device$_\alpha$及應用服務器；反之，重啟金鑰更新程序。

3.2.6 第四階段



圖十二 AppSKey與D-Box更新流程序列圖-5

當接收到確認訊息，end-device$_{\alpha}$（或應用服務器）的動作如下：

1. 將內部儲存器的$AppSKeys_{\alpha\text{-}old}$重置成$AppSKeys_{\alpha\text{-}new}$；
2. 將內部儲存器的D-Box$_{\alpha\text{-}old}$重置成D-Box$_{\alpha\text{-}new}$。

第四章 安全性及功耗分析

在此章節，將先分析LPDEM的安全性以及如何保護系統防禦重送攻擊（replay attack）、竊聽攻擊（eavesdropping attack ）及已知金鑰攻擊（known-key attack）。最後討論LPDEM的功耗。

4.1 安全性評估

4.1.1 LPDEM的安全性

在EDASS算法裡，根據字元的整數值及*dsc*值的總和，將128位元的輸入劃分成16個區塊（16 characters ），且利用檢索表產生密文。換句話說，EDASS中R-Box查表與置換步驟之安全性來自於改變*dsc*的值，例如當兩個相似且差異很小的明文輸入至EDASS進行加密時，會產生完全不同的*dsc*與兩個不同的密文。

在此研究中，DBG算法利用EDASS來產生動態金鑰，最後產生D-Box。由於EDASS對輸入的高敏感性與隨機性，使得駭客難以透過D-Box竊取經DBG算法加密的明文。因此DBG算法既方便計算又能增進D-Box的安全性。

在AES-128系統中，D-Box具有256個元素，因此在SubBytes步驟的加密過程裡，檢索表具256!種可能。如果駭客想解密應用層的訊息，其需要擁有128位元的AppSKey與D-Box，AppSKey與D-Box的組合可能性高達$2^{128}\times 256!$ 種。我們假設駭客利用最先進的技術及設備需要n天（通常是年）來成功的攻擊傳統AES。而在LPDEM系統中，AppSKey與D-Box如前所提，每k天會進行更新，而$n \gg k$。在沒有得知AppSKey與D-Box的前提下，駭客是很難解密已加密的訊息。

4.1.2 已知金鑰攻擊

當駭客知道AppSKey時，即有可能發生已知金鑰攻擊，隨即發現加密機制的運作。在LPDEM系統裡，應用層的訊息已經透過每k天更新AppSKey與D-Box的AES-

128加密，如果駭客獲得了先前的AppSKey，即$AppSKey_{\alpha\text{-}old}$，其仍然無法得知D-Box，因此加密訊息是安全的。此外，在D-Box$_{\alpha\text{-}old}$仍是未知的狀態下，駭客計算出正確的k_{ct}，$AppSKey_{\alpha\text{-}new}$是無法透過$AppSKey_{\alpha\text{-}old}$獲取的。所以可以直言LPDEM系統可以有效地防禦已知金鑰攻擊。

4.1.3 重送攻擊

在LPDEM系統中，時間金鑰k_{ct}是透過網路服務器在AppSKey與D-Box更新程序中的系統次數ct獲得。重送攻擊是駭客複製由網路服務器發送的有效訊息，並假裝成網絡服務器將消息發送到終端設備（或應用服務器），試圖獲取相關訊息。在這樣的情形裡，可能會出現兩種情況。第一個是駭客不進行修改而將原始訊息傳送到終端設備端（或應用程序服務器）。此時，重複傳輸的延遲會使$ct' - ct > \Delta t$，

$ct' - ct \leq \Delta t$將不成立。第二種情形則是駭客修改ct以便$ct' - ct \leq \Delta t$成立。然而，重送的訊息在經過$AppSKey_{\alpha\text{-}new}$與$k_{ct}$計算後會與原傳輸訊息的不同（請參閱AppSKey與D-Box更新程序中，網路服務器收到分別來自終端設備及應用層的$\{ct, Reply_{ED\text{-}\alpha}\}$與$\{ct, Reply_{AP\text{-}\alpha}\}$後的第一與第二步驟）。在更新過程的最後一部分，網路服務器將不會向終端設備和應用服務器發送確認訊息，因此LPDEM系統能夠抵抗重送攻擊。

4.1.4 竊聽攻擊

當駭客從底層網路捕獲大量訊息時，其有機會能成功擷取出重要資料。在LPDEM研究中，網絡服務器發送的AppSKey與D-Box由時間金鑰k_{ct}及前一個的AppSKey（即$AppSKey_{\alpha\text{-}old}$）加密。因此，當$k_{ct}$隨著時間變化，駭客將無法從這些訊息擷取出AppSKey與D-Box。最終，LPDEM系統成功抵禦來自駭客的竊聽攻擊。

4.2 功耗分析

為了分析LPDEM的功耗，我們採用ARM Cortex-M4處理器[67]與低功率結合儲存器（content addressable memory，CAM）分別模擬加密程序與查表過程[68]。Cortex-M4處理器與低功率CAM均採用90nm技術設計，高速搜索為CAM的特色，但同時在處理資料時也消耗非常大量的能源。

在表二中比較AES-128與Simplified-AES數據加密消耗的功率。數據為4個加密步驟的動態功耗，即SubBytes、ShiftRows、MixColumns及AddRoundKey，不考慮處理器的I/O運作，記憶體存取等。在步驟SubBytes與MixColumns中，需要在CAM中檢索S-Box（D-Box），因此很明顯地，比起其他兩個步驟會消耗更多的功率。與AES-128相比，Simplified-AES節省了52.6%（=(3602.9-1708.1)/3602.9）的加密功耗。

表二 AES-128與Simplified-AES的資料加密功耗

Encryption step	AES-128		Simplified-AES	
	Rounds	Power (μW)	Rounds	Power (μW)
AddRoundKey	11	17.3	6	9.5
SubBytes	10	1883.0	5	941.5
ShiftRows	10	7.9	5	3.9
MixColumns	9	1694.7	4	753.2
Total Power		3602.9		1708.1

表三中列出了終端設備一天的功耗。我們假設AppSKey和D-Box每天都會更新，即k=1。終端設備每30分鐘向其網路服務器發送一次數據，即每天48次。為了實現每則訊息的數據完整性，傳統的LoRaWAN和LPDEM都使用帶有NwkSKey的AES-128加密法，即重複10個加密循環。由分析結果得出，與傳統的LoRaWAN相比，LPDEM節省了26.2%（=(345.88 - 255.32)/345.88）的功耗。

表三 AES-128與LPDEM系統的單日功耗

	Traditional LoRaWAN		LPDEM	
	Times	Power (mW)	Times	Power (mW)
Key update	0/day	0.00	1/day	0.40
Data encryption	48/day	172.94	48/day	81.98
Message integrity	48/day	172.94	48/day	172.94
Total Power		345.88		255.32

在LoRaWAN中，從終端設備傳送到應用服務器的訊息都會被加密兩次，其一用於應用層的負載，另一個用於訊息完整性。

由於NwkSKey並不是每k天更新一次，因此可以降低LPDEM用來加密應用層負載過程的功耗，但消息完整性的功耗（倒數第二行）仍然與傳統的LoRaWAN相同。在表2中，可以比較出密鑰更新的功耗遠小於其他部分。如果當k越遠大於1，則可以進一步忽略密鑰更新的功耗。

假定一：

在同是LoRaWAN下，與傳統AES-128相比，LPDEM節省的功率為$(1-\delta)\%$，其中δ為功耗損耗率。

證明：

假定AppSKey與D-Box更新程序的功耗為P_{KU}，且傳統終端裝置加密資料的功耗為P_{EN}。當在只加密五個循環的LPDEM，終端裝置加密資料的功耗為δP_{EN}，其中$0 < \delta < 1$。假定從終端裝置到應用層伺服器資料傳輸的頻率為一天m次，R為功耗節省的百分比，如式4-1所示。

$$R = \left(1 - \frac{m\delta P_{EN} + \frac{1}{k}P_{KU}}{mP_{EN}}\right) \times 100\% \quad \text{(式4-1)}$$

當$\frac{1}{k} \ll m$與$P_{KU} \ll P_{EN}$，金鑰更新的損耗可以被忽略，因此功率節省百分比R可由式4-2所示。

$$R = (1-\delta) \times 100\% \quad \text{(式4-2)}$$

第五章 結論與未來展望

目前在討論物聯網的安全性研究眾多，各式的研究從不同的面向做改善，在本論文中，提出了基於 AES-128 加密法的 LPDEM 系統，用於實現安全性與低功率損失的目標。利用將原固定 S-Box 置換成 D-Box，與週期性地更新終端裝置與應用服務器的加密金鑰（AppSKey）和查詢表（D-Box），能顯著地提高 LoRaWAN 通信的安全級別。此外，將 10 個循環的 AES-128 加密過程減少至 5 個循環，節省加密損耗並延長終端設備的電池壽命。而由分析能得知，LPDEM 能夠節省 26.2％的功耗並抵禦已知密鑰攻擊、重送攻擊及竊聽攻擊。因此能得出結論，LPDEM 是一種具有低功耗特性的安全性方法。有助於保護 LoRaWAN 通信並節省其功耗。

本論文中，僅討論了應用層的數據加密。用於生成 MIC 代碼的 MAC 層的加密密鑰（NwkSKey）並未能定期更新。此外，MIC 代碼生成過程尚未簡化。因此，待未來持續發展用於 NwkSKey 的更新和 MIC 代碼生成的安全且低功耗方法，以便可以進一步最小化 LoRaWAN 中的終端設備的功耗，並且安全性也可以高於當前版本。

參考文獻

[1] T. Robles *et al.*, ‘‘An IoT based reference architecture for smart water management processes,’’ *J. Wireless Mobile Netw., Ubiquitous Comput., Dependable Appl.*, vol. 6, no. 1, pp. 4–23, Mar. 2015.

[2] B. Pokrić *et al.*, ‘‘Augmented reality enabled IoT services for environmental monitoring utilizing serious gaming concept,’’ *J. Wireless Mobile Netw., Ubiquitous Comput., Dependable Appl. (JoWUA)*, vol. 6, no. 1, pp. 37–55, Mar. 2015.

[3] A.-R. Sadeghi, C. Wachsmann, and M. Waidner, ‘‘Security and privacy challenges in industrial Internet of Things,’’ in *Proc. ACM/EDAC/IEEE Design Automat. Conf. (DAC)*, Jun. 2015, pp. 1–6.

[4] S.P. Tseng, B.R. Li, J.L. Pan, and C.J. Lin, ‘‘An application of Internet of Things with motion sensing on smart house,’’ in *Proc. IEEE Int. Conf. Orange Technol. (ICOT)*, Sep. 2014, pp. 65–68.

[5] G. Yang *et al.*, ‘‘A health-IoT platform based on the integration of intelligent packaging, unobtrusive bio-sensor, and intelligent medicine box,’’ *IEEE Trans Ind. Informat.*, vol. 10, no. 4, pp. 2180–2191, Nov. 2014.

[6] M. Ryu, J. Yun, T. Miao, I.-Y. Ahn, S. C. Choi, and J. Kim, ‘‘Designand implementation of a connected farm for smart farming system,’’ in *Proc. IEEE Sensors*, Nov. 2015, pp. 1–4.

[7] J. A. Guerrero-Ibanez, S. Zeadally, and J. Contreras-Castillo, ‘‘Integration challenges of intelligent transportation systems with connected vehicle, cloud computing, and Internet of Things technologies,’’ *IEEE Wireless Commun.*, vol. 22, no. 6, pp. 122–128, Dec. 2015.

[8] P. Cerwall *et al.*, ‘‘Ericsson mobility report,’’ Stockholm, Sweden, Ericsson, Tech. Rep. EAB-17:005964, Jun. 2017.

[9] F. Santoso, M. A. Garratt, S. G. Anavatti, "State-of-the-art intelligent flight control systems in unmanned aerial vehicles", *IEEE Transactions on Automation Science and Engineering*, pp. 1-15, 2017.

[10] E. Husni, G. B. Hertantyo, W. W. Wicaksono, F. C. Hasibuan, A. U. Rahayu, M. A. Triawan, "Applied Internet of Things (IoT): Car monitoring system using IBM BlueMix", *Intelligent Technology and Its Applications (lSITIA)*, pp. 417-422, 2016.

[11] Accessed: Nov. 30, 2018. [Online]. Available: https://yq.aliyun.com/articles/280814

[12] D. Flore, ‘‘3GPP standards for the Internet-of-Things,’’ GSMA MIoT, Huawei, Shenzhen, China, Tech. Rep., Feb. 2016.

[13] Accessed: Apr. 30, 2018. [Online]. Available: https://www.lora- alliance.org/

[14] Accessed: Apr. 30, 2018. [Online]. Available: https://www.sigfox.com/

[15] Accessed: Apr. 30, 2018. [Online]. Available: http://www.weightless.org/

[16] Accessed: Apr. 30, 2018. [Online]. Available: https://www.wi-fi.org/discover-wi-fi/wi-fi-halow/

[17] Accessed: Apr. 30, 2018. [Online]. Available: https://www.ingenu.com/

[18] *Announcing the Advanced Encryption Standard (AES)*, Federal Inf. Process. Standards Publication, United States Nat. Inst. Standards Technol., Nov. 2001.

[19] Y. Chahid, M. Benabdellah, and A. Azizi, ‘‘Internet of Things security,’’ in *Proc. Int. Conf. Wireless Technol., Embedded Intell. Syst. (WITS)*, Apr. 2017, pp. 1–6.

[20] K.L. Tsai, F.Y. Leu, and S.H. Tsai, ‘‘Data encryption method using environmental secret key with server assistance,’’ *Intell. Autom. Soft Comput.*, vol. 22, no. 3, pp. 423–430, Apr. 2016.

[21] P. Prasithsangaree and P. Krishnamurthy, ‘‘Analysis of energy consumption of RC4 and AES algorithms in wireless LANs,’’ in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, Dec. 2003, pp. 1445–1449.

[22] J.M. Kim, H.S. Lee, J. Yi, and M. Park, ‘‘Power adaptive data encryption for energy-efficient and secure communication in solar-powered wireless sensor networks,’’ *J. Sensors*, vol. 2016, Feb. 2016, Art. no. 2678269.

[23] W. Trappe, R. Howard, and R.S. Moore, ‘‘Low-energy security: Limits and opportunities in the Internet of Things,’’ *IEEE Security Privacy*, vol. 13, no. 1, pp. 14–21, Jan. 2015.

[24] K.L. Tsai, M.Y. Ye, and F.Y. Leu, ‘‘Secure power management scheme for WSN,’’ in *Proc. Int. Workshop Manag. Insider Secur. Threats (MIST)*, Oct. 2015, pp. 63–66.

[25] K.L. Tsai, M. Ye, S.H. Tsai, Y.Y. Wang, and Y.H. Zhuang, ‘‘Attack-resistant power management scheme for wireless sensor network,’’ in *Proc. Int. Adv. Robot. Intell. Syst. (ARIS)*, May 2015, pp. 1–4.

[26] B. Cogliati and Y. Seurin, ‘‘Strengthening the known-key security notion for block ciphers,’’ in *Proc. Int. Conf. Fast Softw. Encryption (FSE)*, Mar. 2016, pp. 494–513.

[27] P. Syverson, ‘‘A taxonomy of replay attacks,’’ in *Proc. Comput. Secur. Found. Workshop (CSFW)*, Jun. 1994, pp. 187–191.

[28] J. K. Tugnait, ‘‘Detection of active eavesdropping attack by spoofing relay in multiple antenna systems,’’ *IEEE Wireless Commun. Lett.*, vol. 5, no. 5, pp. 460–463, Oct. 2016.

[29] J. Zhou, Z. Cao, X. Dong, and A. V. Vasilakos, ‘‘Security and privacy for cloud-based IoT: Challenges, countermeasures, and future directions,’’ *IEEE Commun. Mag.*, vol. 55, no. 1, pp. 26–33, Jan. 2017.

[30] Y. Yang, L. Wu, G. Yin, L. Li, H. Zhao, "A survey on security and privacy issues in Internet-of-Things", *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1250-1258, Oct. 2017.

[31] Trustwave Holding Inc., “IoT Cybersecurity Readiness Report”, United States, Nov. 2018.

[32] Accessed: Nov. 30, 2018. [Online]. Available: https://www.eetimes.com/document.asp?doc_id=1333044&page_number=1

[33] H. Ning and H. Liu, ‘‘Cyber-physical-social based security architecture for future Internet of Things,’’ *Adv. Internet Things*, vol. 2, no. 1, pp. 1–7, Jan. 2012.

[34] S. Li, T. Tryfonas, and H. Li, ‘‘The Internet of Things: A security point of view,’’ *Internet Res.*, vol. 26, no. 2, pp. 337–359, Apr. 2016.

[35] M. Horton, L. Chen, and B. Samanta, ‘‘Enhancing the security of IoT enabled robotics: Protecting TurtleBot file system and communication,’’ in *Proc. IEEE Int. Conf. Comput., Netw. Commun. (ICNC)*, Jan. 2017, pp. 1–5.

[36] J. Granjal, E. Monteiro, and J.S. Silva, “Asecureinterconnectionmodel for IPv6 enabled wireless sensor networks,” in *Proc. IFIP Wireless Days*, Venice, Italy, Oct. 2010, pp. 1–6.

[37] S. Sicari, A. Rizzardi, L. Grieco, and A. Coen-Porisini, “Security, pri- vacy and trust in Internet of Things: The road ahead,” *Comput. Netw.*, vol. 76, pp. 146–164, Jan. 2015.

[38] A. Riahi, E. Natalizio, Y. Challal, N. Mitton, and A. Iera, ‘‘Asystemicand cognitive approach for IoT security,’’ in *Proc. IEEE Int. Conf. Comput., Netw. Commun. (ICNC)*, Feb. 2014, pp. 183–188.

[39] T. Heer, O. Garcia-Morchon, R. Hummen, S. L. Keoh, S. S. Kumar, and K. Wehrle, ‘‘Security challenges in the IP-based Internet of Things,’’ *Wireless Pers. Commun.*, vol. 61, no. 3, pp. 527–542, 2011.

[40] R. Kotamsetty and M. Govindarasu, “Adaptive latency-aware query processing on encrypted data for the Internet of Things,” in *Proc. 25th Int. Conf. Comput. Commun. Netw. (ICCCN)*, Aug. 2016, pp. 1–7.

[41] S.A. Salami, J. Baek, K. Salah, and E. Damiani, ‘‘Lightweight encryption for smart home,’’ in *Proc. Int. Conf. Availability, Rel. Secur. (ARES)*, Aug. 2016, pp. 382–388.

[42] D.-H. Bui, D. Puschini, S. Bacles-Min, E. Beigné, and X.-T. Tran, ‘‘Ultra low-power and low-energy 32-bit datapath AES architecture for IoT appli- cations,’’ in *Proc. Int. Conf. IC Des. Technol. (ICICDT)*, Jun. 2016, pp. 1–4.

[43] H. Shafagh, A. Hithnawi, A. Droescher, S. Duquennoy, and W. Hu, “Poster: Towards encrypted query processing for the Internet of Things,” in *Proc. 21st Annu. Int. Conf. Mobile Comput. Netw. (MobiCom)*, Paris, France, 2015, pp. 251–253.

[44] C. E. Weng, V. Sharma, H. C. Chen, and C. H. Mao, ‘‘PEER: Proximity- based energy-efficient routing algorithm for wireless sensor networks,’’ *J. Internet Services Inf. Secur.*, vol. 6, no. 1, pp. 47–56, Feb. 2016.

[45] J. Li, Y. Zhang, X. Chen, and Y. Xiang, ‘‘Secure attribute-based data sharing for resource-limited users in cloud computing,’’ *Comput. Secur.*, vol. 72, pp. 1–12, Jan. 2018.

[46] J. Li, Z. Liu, X. Chen, F. Xhafa, X. Tan, and D. S. Wong, ‘‘L-EncDB: A lightweight framework for privacy-preserving data queries in cloud computing,’’ *Know.lBased Syst.*, vol. 79, pp. 18–26, May 2015.

[47] Z. Cai, H. Yan, P. Li, Z. Huang, and C. Gao,‘‘Towards secure and flexible EHR sharing in mobile health cloud under static assumptions,’’ *Cluster Comput.*, vol. 20, no. 3, pp. 2415–2422, Sep. 2017.

[48] F. Zhang, R. Dojen, and T. Coffey, ‘‘Comparative performance and energy consumption analysis of different AES implementations on a wireless sensor network node,’’ *Int. J. Sensor Netw.*, vol. 10, no. 4, pp. 192–201, 2011.

[49] L. Batina *et al.*, ‘‘Dietary recommendations for lightweight block ciphers: Power, energy and area analysis of recently developed architectures,’’ in *Proc. Int. Workshop Radio Freq. Identification, Secur. Privacy Issues (RFIDSec)*, Nov. 2013, pp. 103–112.

[50] J. C. De Silva, J. J. P. C. Rodrigues, A. M. Alberti, P. Solic, and A. L. L. Aquino, ‘‘LoRaWAN—A low power WAN protocol for Internet of Things: A review and opportunities,’’ in *Proc. Int. Multidisciplinary Conf. Comput. Energy Sci.*, Jul. 2017, pp. 1–6.

[51] D. Bankov, E. Khorov, and A. Lyakhov, ‘‘On the limits of LoRaWAN channel access,’’ in *Proc. Int. Conf. Eng. Telecommun.*, Nov. 2016, pp. 10–14.

[52] K. Mikhaylov, J. Petäjäjärvi, and T. Hänninen, ‘‘Analysisofcapacityand scalability of the LoRa low power wide area network technology,’’ in *Proc. Eur. Wireless Conf.*, May 2016, pp. 119–124.

[53] R. Miller, ‘‘LoRa security—Building a secure LoRa solution,’’ MWR Labs, London, U.K., White Paper, Mar. 2016.

[54] S. Tomasin, S. Zulian, and L. Vangelista, ‘‘Security analysis of LoRaWAN join procedure for Internet of Things networks,’’ in *Proc. Wireless Com- mun. Netw. Conf. Workshops*, Mar. 2017, pp. 1–6.

[55] E. Aras, G. S. Ramachandran, P. Lawrence, and D. Hughes, ‘‘Exploring the security vulnerabilities of LoRa,’’ in *Proc. IEEE Int. Conf. Cybern.*, Jun. 2017, pp. 1–6.

[56] S. Naoui, M.E. Elhdhili, and L.A. Saidane, ‘‘Enhancingthesecurityofthe IoT LoraWAN architecture,’’ in *Proc. Int. Conf. Perform. Eval. Modeling Wired Wireless Netw.*, Nov. 2016, pp. 1–7.

[57] P. Girard. *Low Power Wide Area Networks Security*. Accessed: Apr. 30, 2018. [Online]. Available: https://docbox.etsi.org/workshop/2015/201512_M2MWORKSHOP/S04_WirelessTechnoforIoTandSecurity Challenges/GEMALTO_GIRARD.pdf

[58] J. Kim and J. Song, ‘‘A dual key-based activation scheme for secure LoRaWAN,’’ *Wireless Commun. Mobile Comput.*, vol. 2017, Nov. 2017, Art. no. 6590713.

[59] D. McGrew, ‘‘Low power wireless scenarios and techniques for saving bandwidth without sacrificing security,’’ in *Proc. NIST Lightweight Cryptogr. Workshop*, Jul. 2015, pp. 1–15.

[60] V. L. Dao, V. P. Hoang, A. T. Nguyen, Q. M. Le, "A compact low power AES core on 180nm CMOS process", 2016 International Conference on IC Design and Technology (ICICDT), pp. 1-5, 2016.

[61] Y. L. Huang, F. Y. Leu, P. H. Su, T. H. Sung, and S. C. Liu, ‘‘A secure and high performance wireless sensor network based on symmetric key matrix,’’ in *Proc. Int. Conf. Innov. Mobile Internet Services Ubiquitous Comput.*, Jul. 2016, pp. 470–475.

[62] J.-J. Liu, Y.-L. Huang, F.-Y. Leu, X.-Y. Pan, and L.-R. Chen, ‘‘Generating dynamic box by using an input string,’’ in *Proc. Int. Symp. Mobile Internet Secur.*, Oct. 2017, pp. 1–13.

[63] P. Rogaway, M. Bellare, J. Black and T. Krovetz, “OCB: A Block-cipher Mode of Operation for Efficient Authenticated Encryption,” CCS-8, 2001, pp. 196-205.

[64] G. Manjula and H. S. Mohan, “Constructing Key Dependent Dynamic S-Box for AES Block Cipher System,” iCATccT, 2017, pp. 613-617.

[65] A. Alabaichi and A. I. Salih, “Enhance Security of Advance Encryption Standard Algorithm Based on Key-dependent S-box,” ICDIPC, 2015, pp. 44-53.

[66] S. Arrag, A. Hamdoun, A. Tragha and E. Khamlich Salah, “Implementation of Stronger AES by using Dynamic S-box Dependent of MasterKey,” Journal of Theoretical and Applied Information Technology, vol. 53, no. 2, pp. 196-204, Jul. 2013.

[67] Cortex-M4 Technical Reference Manual, Cambridge, U.K., ARM Ltd., 2009.

[68] K.-L. Tsai, Y.-J. Chang, and Y.-C. Cheng, ‘‘Automatic charge balancing content addressable memory with self-control mechanism,’’ IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 61, no. 10, pp. 2834–2841, Oct. 2014.

[69] K.-L. Tsai, Y.L. Huang, F.Y. Leu, Ilsun You, Y.L. Huang, C.H. Tsai, “AES-128 based Secure Low Power Communication for LoRaWAN IoT Environments.” IEEE Access 2018, vol. 6, 45325–45334, Jul. 2018.